

## **APPENDIX A**

### **RFA SOFTWARE, SOFTWARE SUPPORT, AND QUALIFICATION PROVISIONS**

## **APPENDIX A**

### **RFA SOFTWARE, SOFTWARE SUPPORT, AND QUALIFICATION PROVISIONS**

The following outline presents the top level outline for RFA software, software support, and qualification provisions. In the Hyper Text Markup Language (HTML) version of this appendix, the outline contains hyperlinks to the supporting material appearing in the remainder of the appendix.

#### **RFA Maintenance**

1. Software
  - a. Source Files
  - b. Executable Software
  - c. Segmentation Scripts
2. Design
  - a. Design Decisions
  - b. Architecture (includes Software Component Interaction)
  - c. Environmental Variables
  - d. Database Variables
  - e. External System Interfaces
3. Compilation/Build Procedures
4. Modification Procedures
5. Qualification Provisions

#### **A.1 Software**

The following sections list all the RFA software, executables and segmentation scripts. Figure A.1-1. shows how the RFA software dynamically interrelates.

##### **A.1.1 Source Files**

RFA source code files are delivered on magnetic tape separately from RFA segments. To view files located in /h/RFA/source/, the source files must be loaded into /h/RFA/source/.

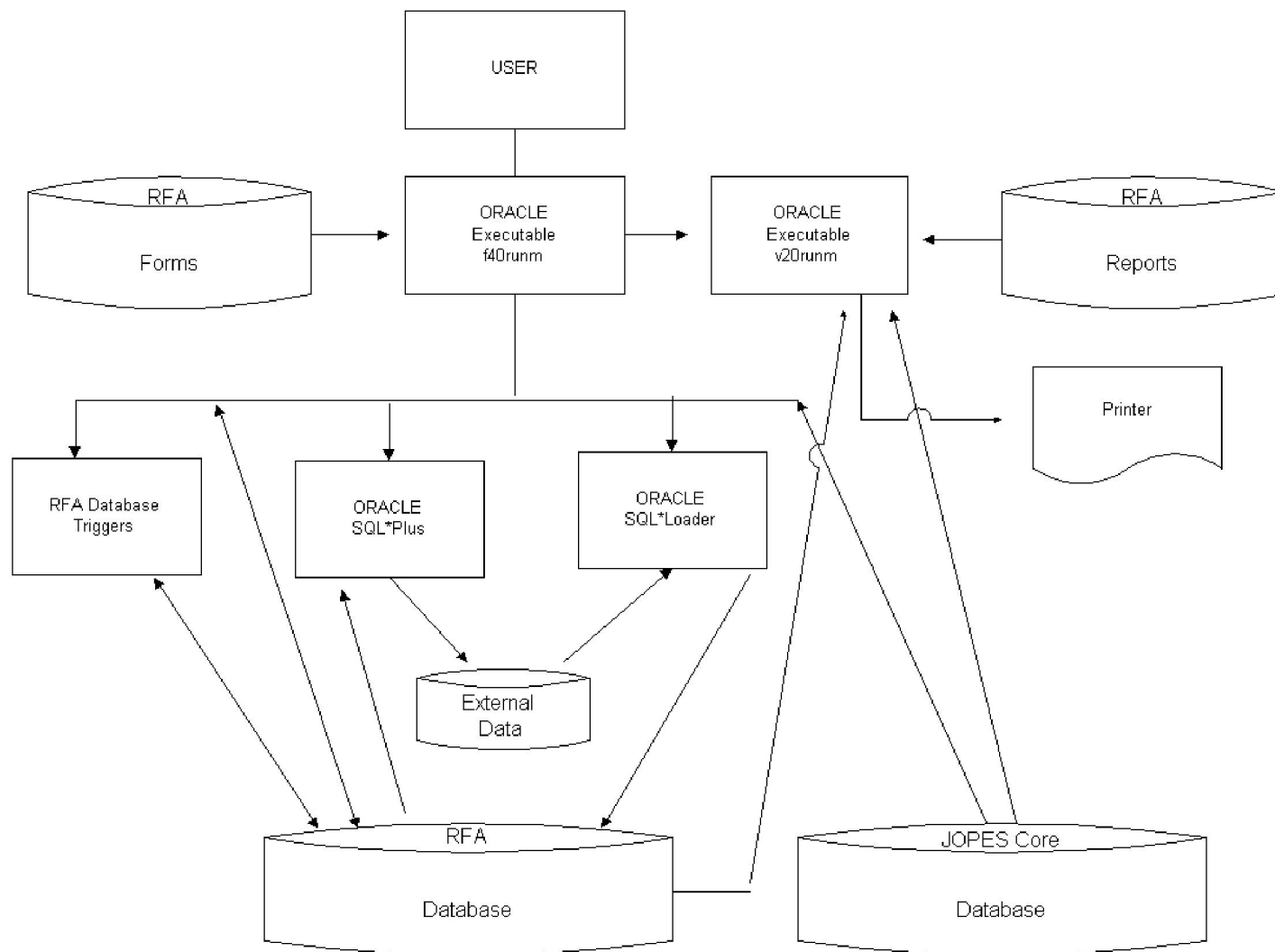


Figure A.1-1 RFA Software Hierarchy

#### **A.1.1.1 ORACLE 4.0 Forms**

The following files are located in the directory /h/RFA/source/:

*acs\_onl.fmb*  
*apo\_bat.fmb*  
*apo\_net.fmb*  
*geo\_del.fmb*  
*geo\_net.fmb*  
*geo\_onl.fmb*  
*help.fmb*  
*help\_onl.fmb*  
*lff\_bat.fmb*  
*lff\_net.fmb*  
*por\_bat.fmb*  
*por\_net.fmb*  
*rfa.fmb*  
*rfa\_ses.fmb*  
*tuc\_bat.fmb*  
*tuc\_del.fmb*  
*tuc\_net.fmb*  
*tuc\_onl.fmb*  
*tud\_bat.fmb*  
*tud\_net.fmb*

#### **A.1.1.2 ORACLE 4.0 Forms Menus**

The following file is located in the directory /h/RFA/source/:

*rfa.mmb*

#### **A.1.1.3 ORACLE 4.0 Forms Libraries**

The following file is located in the directory /h/RFA/source/:

*rfalib.pll*

#### **A.1.1.4 ORACLE 2.0 Reports**

The following files are located in the directory /h/RFA/source/:

*Airport.rdf*  
*Airport\_Aircraft\_Cat.rdf*  
*Airport\_Clearance.rdf*  
*Airport\_Remark.rdf*  
*Apron.rdf*  
*Harbor.rdf*  
*InTransList.rdf*

*Ports\_Of\_Support\_L10.rdf*  
*Priority\_Geo.rdf*  
*Seaport.rdf*  
*Seaport\_Anchorage.rdf*  
*Seaport\_Berth\_Depth.rdf*  
*Seaport\_Cargo\_Capacity.rdf*  
*Seaport\_Channel.rdf*  
*Seaport\_Clearance.rdf*  
*Seaport\_Craft.rdf*  
*Seaport\_Crane.rdf*  
*Seaport\_Floating\_Crane.rdf*  
*Seaport\_Mhe.rdf*  
*Seaport\_Remark.rdf*  
*Seaport\_Standard\_Berth.rdf*  
*Seaport\_Structure.rdf*  
*Service\_Fuel\_L4.rdf*  
*Service\_Fuel\_Rate\_L4.rdf*  
*Service\_Orgn\_Appn\_L11.rdf*  
*Service\_Resupply\_L3.rdf*  
*Service\_Resupply\_Loc\_L2.rdf*  
*Service\_Supply\_L4.rdf*  
*Service\_Supply\_Rate\_L4.rdf*  
*Unit\_Type\_Fuel\_L1.rdf*  
*Unit\_Type\_Fuel\_Rate\_L1.rdf*  
*Unit\_Type\_Supply\_L1.rdf*  
*Unit\_Type\_Supply\_Rate\_L1.rdf*  
*Wharf.rdf*  
*Wharf\_Berth.rdf*  
*Wharf\_Container\_Storage.rdf*  
*Wharf\_Equipment.rdf*  
*aafif\_errors.rdf*  
*aports\_merge.rdf*  
*aports\_xcn.rdf*  
*genRefFile.rdf*  
*geo\_auto\_delete.rdf*  
*geo\_loc\_lst.rdf*  
*geo\_upd\_cycle.rdf*  
*geo\_upd\_session.rdf*  
*jrsEdit.rdf*  
*lff\_xcn.rdf*  
*ports\_dia.rdf*  
*ports\_xcn.rdf*  
*tucha\_auto\_delete.rdf*  
*tucha\_cycle\_update.rdf*  
*tucha\_list.rdf*  
*tucha\_sess.rdf*  
*tucha\_xcn.rdf*  
*tudet\_lst.rdf*

*tudet\_sess.rdf*  
*tudet\_upd.rdf*  
*tudet\_xcn.rdf*

#### **A.1.1.5 ORACLE Reports 2.0 Libraries**

The following files are located in the directory /h/RFA/rfa\_home/:

*genRpts.pll*  
*rfa\_report.pll*  
*rfalib.pll*

#### **A.1.1.6 ORACLE SQL\*Loader Libraries**

The following files are located in the directory /h/RFA/rfa\_home/:

*apoload.ctl*  
*cnoload.ctl*  
*di2load.ctl*  
*di3load.ctl*  
*di4load.ctl*  
*di5load.ctl*  
*di6load.ctl*  
*di7load.ctl*  
*di8load.ctl*  
*diaload.ctl*  
*lffload.ctl*  
*tucload.ctl*  
*tudload.ctl*

#### **A.1.1.7 ORACLE SQL Scripts**

The following files are located in the directory /h/RFA/rfa\_home/:

*spool\_jrs.sql*  
*spool\_sql.sql*

#### **A.1.2 Executable Software**

RFA has no executable files of its own, but uses ORACLE executables *f40runm* to execute forms and *r20runm* to run reports. These executables use the following object files that are located in the directory /h/RFA/rfa\_home/:

- a. ORACLE 4.0 Forms:

*acs\_onl.fmx*  
*apo\_bat.fmx*

*apo\_net.fmx*  
*geo\_del.fmx*  
*geo\_net.fmx*  
*geo\_onl.fmx*  
*help.fmx*  
*help\_onl.fmx*  
*lff\_bat.fmx*  
*lff\_net.fmx*  
*por\_bat.fmx*  
*por\_net.fmx*  
*rfa.fmx*  
*rfa\_ses.fmx*  
*tuc\_bat.fmx*  
*tuc\_del.fmx*  
*tuc\_net.fmx*  
*tuc\_onl.fmx*  
*tud\_bat.fmx*  
*tud\_net.fmx*

b. ORACLE 4.0 Forms Menu:

*rfa.mmx*

c. ORACLE 4.0 Forms Library:

*rfalib.lib*

d. ORACLE 2.0 Reports:

*Airport.rep*  
*Airport\_Aircraft\_Cat.rep*  
*Airport\_Clearance.rep*  
*Airport\_Remark.rep*  
*Apron.rep*  
*Harbor.rep*  
*InTransitList.rep*  
*Ports\_Of\_Support\_l10.rep*  
*Priority\_Geo.rep*  
*Seaport.rep*  
*Seaport\_Anchorage.rep*  
*Seaport\_Berth\_Depth.rep*  
*Seaport\_Cargo\_Capacity.rep*  
*Seaport\_Channel.rep*  
*Seaport\_Clearance.rep*  
*Seaport\_Craft.rep*  
*Seaport\_Crane.rep*  
*Seaport\_Floating\_Crane.rep*  
*Seaport\_Mhe.rep*

*Seaport\_Remark.rep*  
*Seaport\_Standard\_Berth.rep*  
*Seaport\_Structure.rep*  
*Service\_Fuel\_L4.rep*  
*Service\_Fuel\_Rate\_L4.rep*  
*Service\_Orgn\_Appn\_L1.rep*  
*Service\_Resupply\_L3.rep*  
*Service\_Resupply\_Loc\_L2.rep*  
*Service\_Supply\_L4.rep*  
*Service\_Supply\_Rate\_L4.rep*  
*Unit\_Type\_Fuel\_L1.rep*  
*Unit\_Type\_Fuel\_Rate\_L1.rep*  
*Unit\_Type\_Supply\_L1.rep*  
*Unit\_Type\_Supply\_Rate\_L1.rep*  
*Wharf.rep*  
*Wharf\_Berth.rep*  
*Wharf\_Container\_Storage.rep*  
*Wharf\_Equipment.rep*  
*aafif\_errors.rep*  
*aports\_merge.rep*  
*aports\_xcn.rep*  
*genRefFile.rep*  
*geo\_auto\_delete.rep*  
*geo\_loc\_lst.rep*  
*geo\_upd\_cycle.rep*  
*geo\_upd\_session.rep*  
*jrsEdit.rep*  
*lff\_xcn.rep*  
*ports\_dia.rep*  
*ports\_xcn.rep*  
*tucha\_auto\_delete.rep*  
*tucha\_cycle\_update.rep*  
*tucha\_list.rep*  
*tucha\_sess.rep*  
*tucha\_xcn.rep*  
*tucha\_lst.rep*  
*tucha\_sess.rep*  
*tudet\_upd.rep*  
*tudet\_xcn.rep*

### **A.1.3 Segmentation Scripts**

RFA has two segments, a database segment and a client segment. The database segment is called “*RFADB*” and is located in /h/RFADB/ while the client segment is called “*RFA*” and is located in /h/RFA/.



### A.1.3.1 RFADB Segment

- a. /h/RFADB/SegDescrip/ holds the segment's environment specification files:

*DEINSTALL*  
*Data*  
*Hardware*  
*ModName*  
*PostInstall*  
*ReleaseNotes*  
*Requires*  
*Security*  
*SegType*  
*VERSION*  
*Validated*

- b. /h/RFADB/Scripts/ holds commonly used UNIX scripts:

*determine\_oracle\_password*  
*dir.csh*  
*enter\_new\_oracle\_password*  
*pass.csh*  
*warning.csh*

- c. /h/RFADB/install/ contains UNIX and SQL scripts which are used in granting and revoking RFA privileges to users:

*check\_oracle\_user.sql*  
*check\_rfa\_user.sql*  
*delete.sql*  
*drop\_syn.csh*  
*insert.sql*  
*make\_drop\_syn.sql*  
*make\_syn.sql*  
*new\_rfa\_user*  
*revoke\_rfa\_user*  
*revoke\_user.sql*  
*syn.csh*  
*tm\_syn.sql*  
*user.sql*

- d. /h/RFADB/data/ holds unclassified test data files to evaluate the RFA system:

*apo.exp*  
*apo\_geo.dat*  
*dummy*  
*errors.dat*  
*geo\_cord.dat*

*geo\_geo.dat*  
*geo\_uic.dat*  
*inp\_apo.par*  
*inp\_help.par*  
*inp\_lff.par*  
*inp\_por\_par*  
*jrs\_crit.dat*  
*lff.exp* (binary ORACLE export file, cannot view)  
*lprcheck.dat*  
*online\_help.exp* (binary ORACLE export file, cannot view)  
*por.exp* (binary ORACLE export file, cannot view)  
*tucha.dat*  
*tudet.dat*

- e. /h/RFADB/sql/ holds the SQL and SQL\*Loader scripts required to install the RFADB segment:

*af\_comp.sql*  
*alter\_user.sql*  
*apo\_batb.sql*  
*apo\_bats.sql*  
*apo\_calc.sql*  
*apo\_conx.sql*  
*apo\_edtb.sql*  
*apo\_edts.sql*  
*apo\_geo.ctl*  
*apo\_load.sql*  
*apo\_tabs.sql*  
*apo\_tgrs.sql*  
*comp\_gr.sql*  
*create\_rfa\_analyze.sql*  
*deinstall.sql*  
*droptemp.sql*  
*errors.ctl*  
*gen\_load.sql*  
*gen\_pack.sql*  
*gen\_tabs.sql*  
*geo\_conx.sql*  
*geo\_cord.ctl*  
*geo\_edtb.sql*  
*geo\_edts.sql*  
*geo\_geo.ctl*  
*geo\_tabs.sql*  
*geo\_tgrs.sql*  
*geo\_uic.ctl*  
*jrs\_crit.ctl*  
*lff\_bods.sql*  
*lff\_bulb.sql*

*lff\_buls.sql*  
*lff\_conx.sql*  
*lff\_spcs.sql*  
*lff\_tabs.sql*  
*lff\_tgrs.sql*  
*lprcheck.ctl*  
*por\_basp.sql*  
*por\_bd1b.sql*  
*por\_bd2b.sql*  
*por\_bd3b.sql*  
*por\_bd4b.sql*  
*por\_bt1b.sql*  
*por\_bt2b.sql*  
*por\_bt3b.sql*  
*por\_bt4b.sql*  
*por\_btpb.sql*  
*por\_cnob.sql*  
*por\_conx.sql*  
*por\_edb1.sql*  
*por\_edb2.sql*  
*por\_edb3.sql*  
*por\_edb4.sql*  
*por\_edsp.sql*  
*por\_popb.sql*  
*por\_pope.sql*  
*por\_pops.sql*  
*por\_preb.sql*  
*por\_roll.sql*  
*por\_tabs.sql*  
*por\_tgrs.sql*  
*rfa\_role.sql*  
*rfa\_ts.sql*  
*rfa\_user.sql*  
*role.sql*  
*t\_m\_gr.sql*  
*t\_m\_role.sql*  
*temp.sql*  
*tuc\_conx.sql*  
*tuc\_pack.sql*  
*tuc\_rpts.sql*  
*tuc\_sra.ctl*  
*tuc\_tabs.sql*  
*tuc\_tgrs.sql*  
*tud\_conx.sql*  
*tud\_pack.sql*  
*tud\_sra.ctl*  
*tud\_tabs.sql*  
*tud\_tgrs.sql*

*val\_load.sql*  
*val\_tabs.sql*

#### **A.1.3.2 RFA Segment**

- a. /h/RFA/SegDescrip/ holds the segment's environment specification files:

*DEINSTALL*  
*Hardware*  
*ModName*  
*PostInstall*  
*ReleaseNotes*  
*Requires*  
*Security*  
*SegType*  
*VERSION*  
*Validated*

- b. /h/RFA/Scripts/ contains the commonly used UNIX Script:

*pass.csh*

- c. /h/RFA/data/Profiles/ contains Executive Manager profile files required by the GCCS desktop:

*LaunchDesc.RFA*  
*LaunchList.RFA*  
*Profiles.RFA*  
*rfa.img*

- d. /h/RFA/ contains the environment variables assignment file:

*rfaenv*

- e. /h/RFA/prog/ contains the RFA launch scripts:

*RFA\_launch*  
*RFA\_forms*

## **A.2 “As Built” Design**

### **A.2.1 Design Decisions**

RFA is the migration of the COBOL Reference File Maintenance Subsystem of the Joint Operation Planning System (JOPS) to a SUN UNIX client server environment. RFA is designed to use ORACLE Forms and ORACLE Reports against an ORACLE database. Transaction files in JRS format must be present in the subdirectory /h/RFA/data/rfa\_net. When the RFA user makes changes to the reference files, the changes are made against local files, not the “live” reference files being used by other GCCS users. As changes are made, RFA software captures all changes by the use of ORACLE database triggers. These “before” and “after” images are reported to the user to verify the changes made. RFA consolidates all changes to the smallest set and generates the differences between the local reference file and the “live” reference file. These differences are described in an ORACLE SQL update file that can be executed on the JOPES database server to update the reference file tables.

### **A.2.2 Architecture**

The RFA architecture can be described in a hierarchical fashion. Each element in the hierarchy consists of a set of physical files. The highest elements are termed work packages. Work packages themselves consist of logical units which are logical collections of physical files.

As shown below in Figure A.2.2-1, the RFA work packages are GEO, TUCHA, TUDET, LFF, PORTS, APORTS, and MISCELLANEOUS.

The identification of the elements in each logical unit is depicted in figures in the discussion of each work package.

The identification of the elements in the lowest level in the RFA hierarchy, the physical file, are listed alphabetically by work package by logical unit. For example, *geo\_cord.ctl* is the sixth physical file unit of GEO Initialization, the first logical unit of the GEO work package.

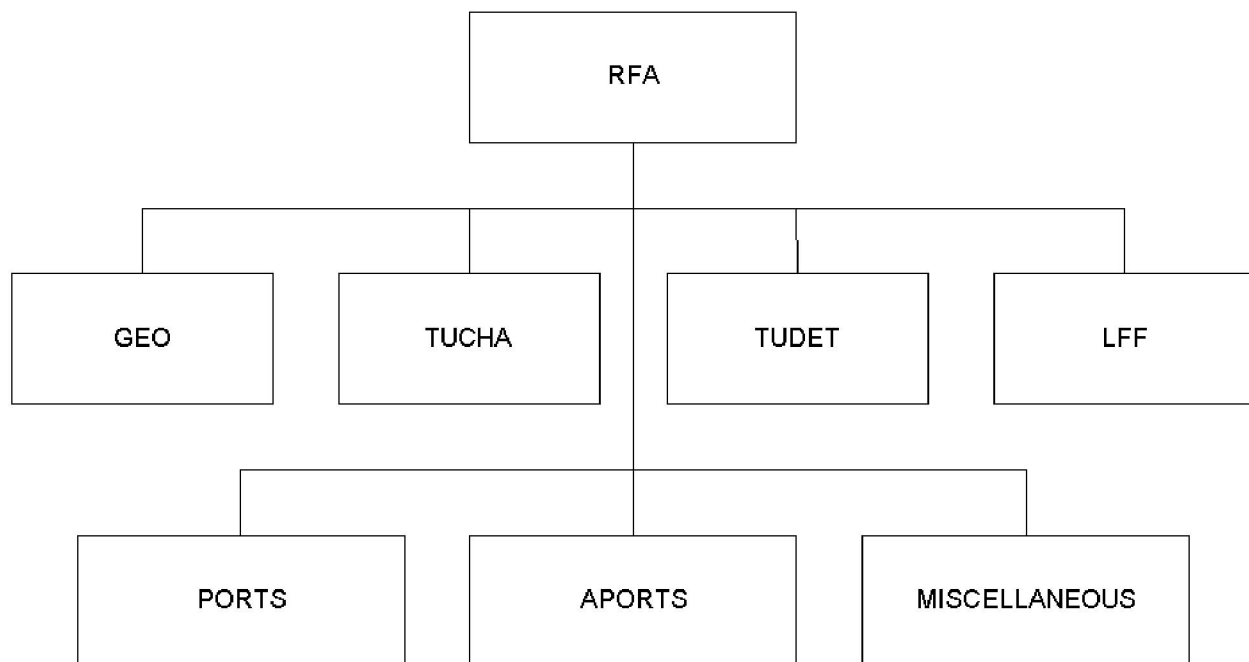


Figure A.2.2-1. RFA Work Packages

#### A.2.2.1 GEO Work Package

The GEO work package has four logical units: GEO Initialization, GEO On-line Update, GEO Network, and GEO Reports as shown in Figure A.2.2.1-1.

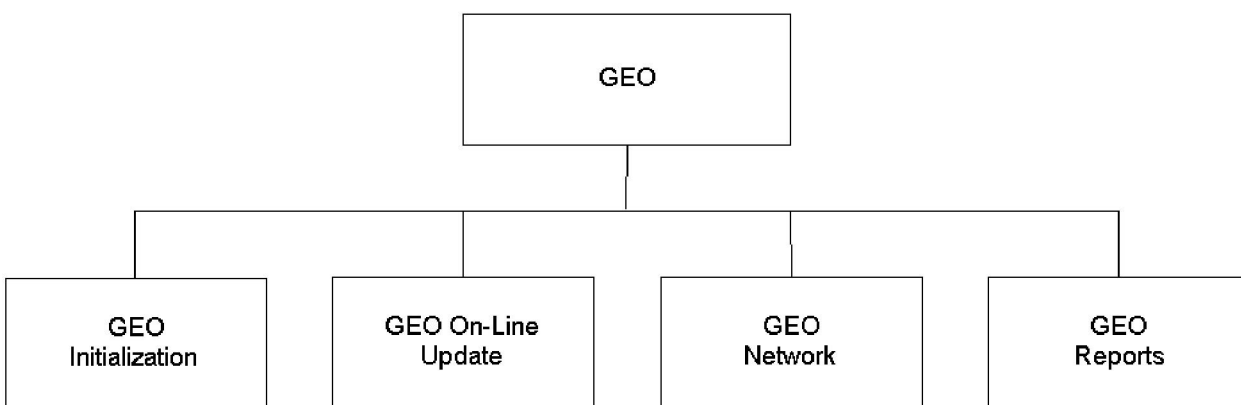


Figure A.2.2.1-1. GEO Work Package

#### **A.2.2.1.1 GEO Initialization**

The GEO Initialization logical unit enables RFA installers to create GEO database objects and grant required privileges to those objects. The GEO Initialization unit consists of the following files in the indicated directories:

```
lprcheck.ctl (/h/RFADB/sql/)
lprcheck.dat (/h/RFADB/data/)
geo_conx.sql (/h/RFADB/sql/)
geo_geo.ctl (/h/RFADB/sql/)
geo_geo.dat (/h/RFADB/data/)
geo_cord.ctl (/h/RFADB/sql/)
geo_cord.dat (/h/RFADB/data/)
geo_tabs.sql (/h/RFADB/sql/)
geo_tgrs.sql (/h/RFADB/sql/)
geo_uic.ctl (/h/RFADB/sql/)
geo_uic.dat (/h/RFADB/data/)
```

#### **A.2.2.1.2 GEO On-line Update**

The GEO On-line Update logical unit enables RFA users to make interactive changes to the GEOGRAPHIC\_LOCATION table. Database triggers capture “before” and “after” images. A GEO autodelete function is also included. The GEO On-line Update unit consists of the following files in the indicated directories:

```
geo_del.fmb (/h/RFA/source/)
geo_onl.fmb (/h/RFA/source/)
```

#### **A.2.2.1.3 GEO Network**

The GEO Network logical unit enables RFA users to reduce all changes to the GEO file and generate a SQL script file of differences between the local GEOGRAPHIC\_LOCATION table and the “live” table. The GEO Network unit consists of the following file in the indicated directory:

```
geo_net.fmb (/h/RFA/source/)
```

#### **A.2.2.1.4 GEO Reports**

The GEO Reports logical unit enables RFA users to generate GEO reports. The GEO Reports unit consists of the following files in the indicated directories:

```
geo_auto_delete.rdf (/h/RFA/source/)
geo_loc_1st.rdf (/h/RFA/source/)
geo_upd_cycle.rdf (/h/RFA/source/)
geo_upd_session.rdf (/h/RFA/source/)
```

### A.2.2.2 TUCHA Work Package

The TUCHA work package has five logical units: TUCHA Initialization, TUCHA Batch Update, TUCHA On-line Update, TUCHA Network, and TUCHA Reports as shown in Figure A.2.2.2-1.

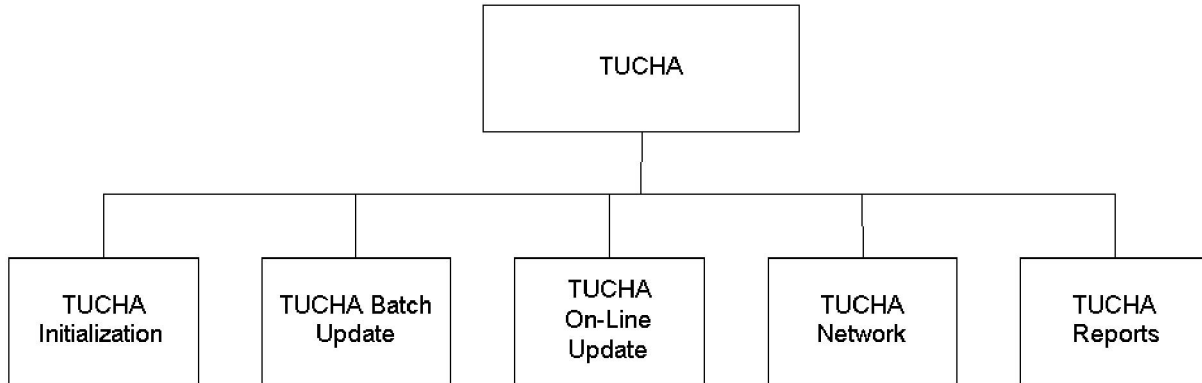


Figure A.2.2.2-1. TUCHA Work Package

#### A.2.2.2.1 TUCHA Initialization

The TUCHA Initialization logical unit enables RFA installers to create TUCHA database objects and grant required privileges to those objects. The TUCHA Initialization unit consists of the following files in the indicated directories:

*tuc\_conx.sql* (/h/RFADB/sql/)  
*tuc\_pack.sql* (/h/RFADB/sql/)  
*tuc\_rpts.sql* (/h/RFADB/sql/)  
*tuc\_sra.ctl* (/h/RFADB/sql/)  
*tuc\_tabs.sql* (/h/RFADB/sql/)  
*tuc\_tgrs.sql* (/h/RFADB/sql/)  
*tucha.dat* (/h/RFADB/data/)

#### A.2.2.2.2 TUCHA Batch Update

The TUCHA Batch Update logical unit enables RFA users to process transactions files submitted in accordance with the JRS for TUCHA. Database triggers capture “before” and “after” images to all TUCHA tables. The TUCHA Batch Update unit consists of the following files in the indicated directories:

*tuc\_bat.fmb* (/h/RFA/source/)  
*tucload.ctl* (/h/RFA/rfa\_home/)



#### A.2.2.2.3 TUCHA On-line Update

The TUCHA On-line logical unit enables the RFA user to delete canceled TUCHA entries via an autodelete function. The TUCHA On-line unit consists of the following files in the indicated directories:

*tuc\_del.fmb* (/h/RFA/source/)  
*tuc\_onl.fmb* (/h/RFA/source/)

#### A.2.2.2.4 TUCHA Network

The TUCHA Network logical unit enables RFA users to reduce all changes to TUCHA tables and generate a SQL script file of differences between the local TUCHA tables and the “live” tables. The TUCHA Network unit consists of the following file in the indicated directory:

*tuc\_net.fmb* (/h/RFA/source/)

#### A.2.2.2.5 TUCHA Reports

The TUCHA Reports logical unit enables RFA users to generate TUCHA Reports. The TUCHA Reports unit consists of the following files in the indicated directory:

*tucha\_auto\_delete.rdf* (/h/RFA/source/)  
*tucha\_cycle\_update.rdf* (/h/RFA/source/)  
*tucha\_list.rdf* (/h/RFA/source/)  
*tucha\_sess.rdf* (/h/RFA/source/)  
*tucha\_xcn.rdf* (/h/RFA/source/)

#### A.2.2.3 TUDET Work Package

The TUDET work package has four logical units: TUDET Initialization, TUDET Batch Update, TUDET Network, and TUDET Reports as shown in Figure A.2.2.3-1.

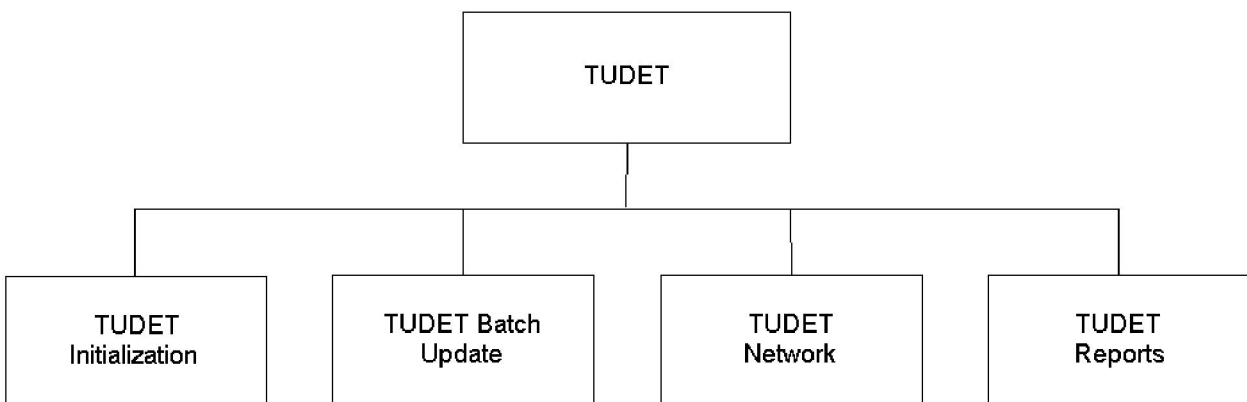


Figure A.2.2.3-1. TUDET Work Package

#### **A.2.2.3.1 TUDET Initialization**

The TUDET Initialization logical unit enables RFA installers to create TUDET database objects and grant required privileges to those objects. The TUDET Initialization unit consist of the following files in the indicated directories:

```
tud_conx.sql (/h/RFADB/sql/)
tud_pack.sql (/h/RFADB/sql/)
tud_sra.ctl (/h/RFADB/sql/)
tud_tabs.sql (/h/RFADB/sql/)
tud_tgrs.sql (/h/RFADB/sql/)
tudet.dat (/h/RFADB/data/)
```

#### **A.2.2.3.2 TUDET Batch Update**

The TUDET Batch Update logical unit enables RFA users to process transactions files submitted in accordance with the JRS for TUDET. Database triggers capture “before” and “after” images for changes to the TUDET table. The TUDET Batch Update unit consists of the following files in the indicated directories:

```
tud_bat.fmb (/h/RFA/source/)
tudload.ctl (/h/RFA/rfa_home/)
```

#### **A.2.2.3.3 TUDET Network**

The TUDET Network logical unit enables RFA users to reduce all changes to TUDET tables and generate a SQL script file of differences between the local TUDET table and the “live” table. The TUDET Network unit consists of the following file in the indicated directory:

```
tud_net.fmb (/h/RFA/source/)
```

#### **A.2.2.3.4 TUDET Reports**

The TUDET Reports logical unit enables RFA users to generate TUDET reports. The TUDET Reports unit consists of the following files in the indicated directories:

```
tudet_lst.rdf (/h/RFA/source/)
tudet_sess.rdf (/h/RFA/source/)
tudet_upd.rdf (/h/RFA/source/)
tudet_xcn.rdf (/h/RFA/source/)
```

#### **A.2.2.4 LFF Work Package**

The LFF Work Package has four logical units: LFF Initialization, LFF Batch Update, LFF Network, and LFF Reports as shown in Figure A.2.2.4-1.

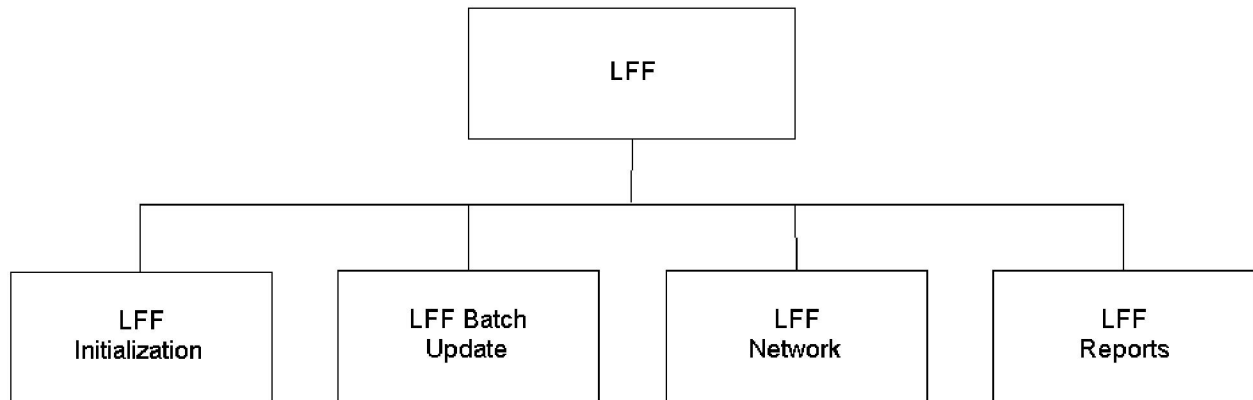


Figure A.2.2.4-1. LFF Work Package

#### A.2.2.4.1 LFF Initialization

The LFF Initialization logical unit enables RFA installers to create LFF database objects and grant required privileges to those objects. The LFF Initialization unit consist of the following files in the indicated directories:

*lff\_bods.sql* (/h/RFADB/sql/)  
*lff\_bulb.sql* (/h/RFADB/sql/)  
*lff\_buls.sql* (/h/RFADB/sql/)  
*lff\_conx.sql* (/h/RFADB/sql/)  
*lff.exp* (/h/RFADB/data/) (binary ORACLE export file, cannot view)  
*lff\_spcs.sql* (/h/RFADB/sql/)  
*lff\_tabs.sql* (/h/RFADB/sql/)  
*lff\_tgrs.sql* (/h/RFADB/sql/)

#### A.2.2.4.2 LFF Batch Update

The LFF Batch Update logical unit enables RFA users to process transactions files submitted in accordance with the JRS for LFF. Database triggers capture “before” and “after” images for changes to the LFF tables. The LFF Batch Update unit consists of the following files in the indicated directories:

*lff\_bat.fmb* (/h/RFA/source/)  
*lffload.ctl* (/h/RFA/rfa\_home/)

#### A.2.2.4.3 LFF Network

The LFF Network logical unit enables RFA users to reduce all changes to LFF tables and generate a SQL script file of differences between the local LFF tables and the “live” tables. The LFF Network unit consists of the following files in the indicated directory:

*lff\_net.fmb* (/h/RFA/source/)

#### A.2.2.4.4 LFF Reports

The LFF Reports logical unit enables RFA users to generate LFF reports. The LFF Reports unit consists of the following files in the indicated directories:

*Ports\_Of\_Support\_L10.rdf* (/h/RFA/source/)  
*Service\_Fuel\_L4.rdf* (/h/RFA/source/)  
*Service\_Fuel\_Rate\_L4.rdf* (/h/RFA/source/)  
*Service\_Orgn\_Appn\_L11.rdf* (/h/RFA/source/)  
*Service\_Resupply\_L3.rdf* (/h/RFA/source/)  
*Service\_Resupply\_Loc\_L2.rdf* (/h/RFA/source/)  
*Service\_Supply\_L4.rdf* (/h/RFA/source/)  
*Service\_Supply\_Rate\_L4.rdf* (/h/RFA/source/)  
*Unit\_Type\_Fuel\_L1.rdf* (/h/RFA/source/)  
*Unit\_Type\_Fuel\_Rate\_L1.rdf* (/h/RFA/source/)  
*Unit\_Type\_Supply\_L1.rdf* (/h/RFA/source/)  
*Unit\_Type\_Supply\_Rate\_L1.rdf* (/h/RFA/source/)  
*lff\_xcn.rdf* (/h/RFA/source/)

#### A.2.2.5 PORTS Work Package

The PORTS work package has four logical units: PORTS Initialization, PORTS Batch Update, PORTS Network, and PORTS Reports as shown in Figure A.2.2.5-1.

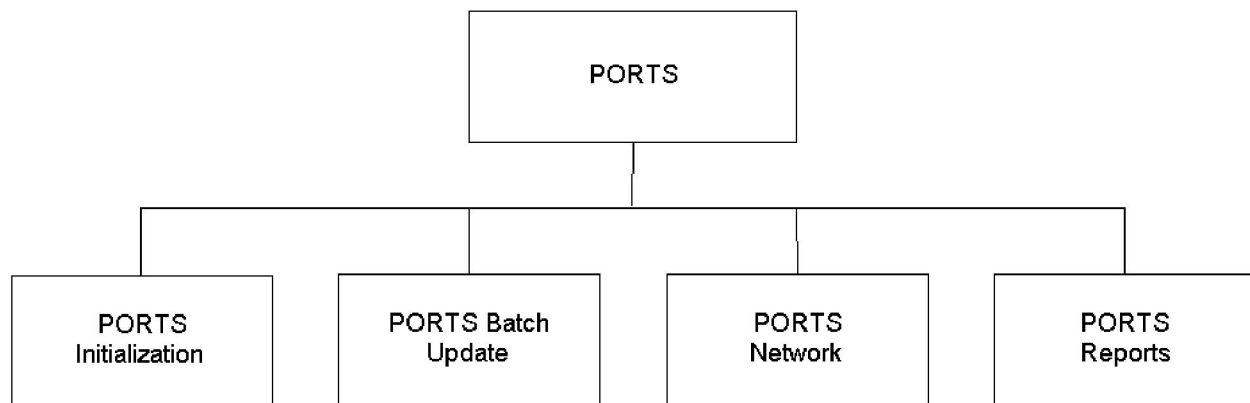


Figure A.2.2.5-1. PORTS Work Package

##### A.2.2.5.1 PORTS Initialization

The PORTS Initialization logical unit enables RFA installers to create PORTS database objects and grant required privileges to those objects. The PORTS Initialization unit consists of the following files in the indicated directories:

*inp\_por.par* (/h/RFADB/data/)  
*por\_basp.sql* (/h/RFADB/sql/)  
*por\_bd1b.sql* (/h/RFADB/sql/)

*por\_bd2b.sql* (/h/RFADB/sql/)  
*por\_bd3b.sql* (/h/RFADB/sql/)  
*por\_bd4b.sql* (/h/RFADB/sql/)  
*por\_bt1b.sql* (/h/RFADB/sql/)  
*por\_bt2b.sql* (/h/RFADB/sql/)  
*por\_bt3b.sql* (/h/RFADB/sql/)  
*por\_bt4b.sql* (/h/RFADB/sql/)  
*por\_btpb.sql* (/h/RFADB/sql/)  
*por\_cnob.sql* (/h/RFADB/sql/)  
*por\_conx.sql* (/h/RFADB/sql/)  
*por\_edb1.sql* (/h/RFADB/sql/)  
*por\_edb2.sql* (/h/RFADB/sql/)  
*por\_edb3.sql* (/h/RFADB/sql/)  
*por\_edb4.sql* (/h/RFADB/sql/)  
*por\_edsp.sql* (/h/RFADB/sql/)  
*por.exp* (/h/RFADB/data/) (binary ORACLE export file, cannot view)  
*por\_popb.sql* (/h/RFADB/sql/)  
*por\_pops.sql* (/h/RFADB/sql/)  
*por\_pope.sql* (/h/RFADB/sql/)  
*por\_preb.sql* (/h/RFADB/sql/)  
*por\_roll.sql* (/h/RFADB/sql/)  
*por\_tabs.sql* (/h/RFADB/sql/)  
*por\_tgrs.sql* (/h/RFADB/sql/)

#### **A.2.2.5.2 PORTS Batch Update**

The PORTS Batch Update logical unit enables RFA users to process transactions files submitted in accordance with the JRS for PORTS. Database triggers capture “before” and “after” images for changes to the PORTS tables. The PORTS Batch Update unit consists of the following files in the indicated directories:

*cnoload.ctl* (/h/RFA/rfa\_home/)  
*di2load.ctl* (/h/RFA/rfa\_home/)  
*di3load.ctl* (/h/RFA/rfa\_home/)  
*di4load.ctl* (/h/RFA/rfa\_home/)  
*di5load.ctl* (/h/RFA/rfa\_home/)  
*di6load.ctl* (/h/RFA/rfa\_home/)  
*di7load.ctl* (/h/RFA/rfa\_home/)  
*di8load.ctl* (/h/RFA/rfa\_home/)  
*diaload.ctl* (/h/RFA/rfa\_home/)  
*por\_bat.fmb* (/h/RFA/source/)

### **A.2.2.5.3 PORTS Network**

The PORTS Network logical unit enables RFA users to reduce all changes to PORTS tables and generate a SQL script file of differences between the local PORTS tables and the “live” tables. The PORTS Network unit consists of the following file in the indicated directory:

*por\_net.fmb* (/h/RFA/source/)

### **A.2.2.5.4 PORTS Reports**

The PORTS Reports logical unit enables RFA users to generate PORTS Reports. The PORTS Reports unit consists of the following files in the indicated directory:

*Harbor.rdf* (/h/RFA/source/)  
*Seaport.rdf* (/h/RFA/source/)  
*Seaport\_Anchorage.rdf* (/h/RFA/source/)  
*Seaport\_Berth\_Depth.rdf* (/h/RFA/source/)  
*Seaport\_Cargo\_Capacity.rdf* (/h/RFA/source/)  
*Seaport\_Channel.rdf* (/h/RFA/source/)  
*Seaport\_Clearance.rdf* (/h/RFA/source/)  
*Seaport\_Craft.rdf* (/h/RFA/source/)  
*Seaport\_Crane.rdf* (/h/RFA/source/)  
*Seaport\_Floating\_Crane.rdf* (/h/RFA/source/)  
*Seaport\_Mhe.rdf* (/h/RFA/source/)  
*Seaport\_Remark.rdf* (/h/RFA/source/)  
*Seaport\_Standard\_Berth.rdf* (/h/RFA/source/)  
*Seaport\_Structure.rdf* (/h/RFA/source/)  
*Wharf.rdf* (/h/RFA/source/)  
*Wharf\_Berth.rdf* (/h/RFA/source/)  
*Wharf\_Container\_Storage.rdf* (/h/RFA/source/)  
*Wharf\_Equipment.rdf* (/h/RFA/source/)  
*ports\_dia.rdf* (/h/RFA/source/)  
*ports\_xcn.rdf* (/h/RFA/source/)

### **A.2.2.6 APORTS Work Package**

The APORTS work package has four logical units: APORTS Initialization, APORTS Batch Update, APORTS Network, and APORTS Reports as shown in Figure A.2.2.6-1.

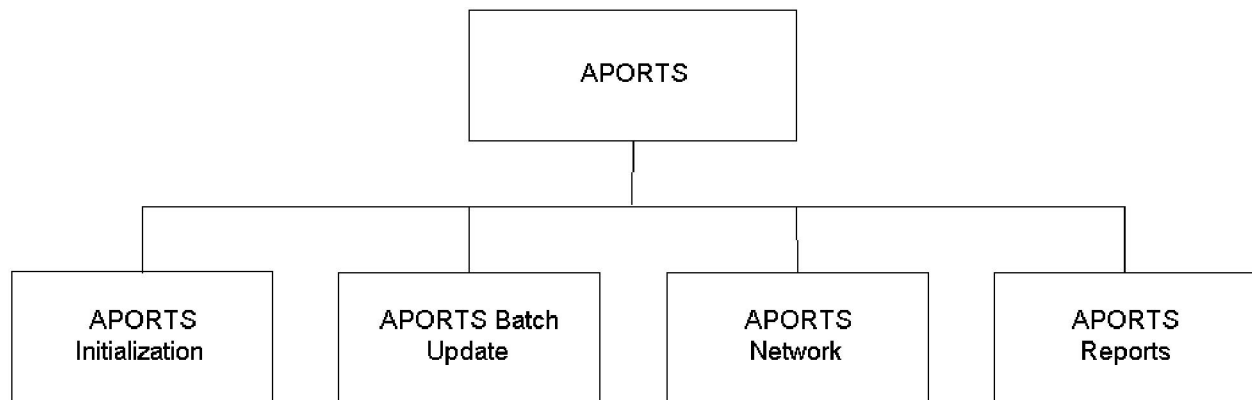


Figure A.2.2.6-1. PORTS Work Package

#### A.2.2.6.1 APORTS Initialization

The APORTS Initialization logical unit enables RFA installers to create APORTS database objects and grant required privileges to those objects. The APORTS Initialization unit consists of the following files in the indicated directories:

```

apo_batb.sql (/h/RFADB/sql/)
apo_bats.sql (/h/RFADB/sql/)
apo_calc.sql (/h/RFADB/sql/)
apo_conx.sql (/h/RFADB/sql/)
apo_edtb.sql (/h/RFADB/sql/)
apo_edts.sql (/h/RFADB/sql/)
apo.exp (/h/RFADB/data/)
apo_load.sql (/h/RFADB/sql/)
apo_geo.ctl (/h/RFADB/sql/)
apo_tabs.sql (/h/RFADB/sql/)
apo_tgrs.sql (/h/RFADB/sql/)
inp_apo.par (/h/RFADB/data/)

```

#### A.2.2.6.2 APORTS Batch Update

The APORTS Batch Update logical unit enables RFA users to process transactions files submitted in accordance with the Joint Reporting Structure for APORTS. Database triggers capture “before” and “after” images for changes to the APORTS tables. The APORTS Batch Update unit consists of the following files in the indicated directory:

```

apo_bat.fmb (/h/RFA/source/)
apoload.ctl (/h/RFA/source/)

```

#### A.2.2.6.3 APORTS Network

The APORTS Network logical unit enables RFA users to reduce all changes to APORTS tables and generate a SQL script file of differences between the local APORTS tables and the “live” tables. The APORTS

Network unit consists of the following file in the indicated directory:

*apo\_net.fmb* (/h/RFA/source/)

#### A.2.2.6.4 APORTS Reports

The APORTS Reports logical unit enables RFA users to generate APORTS Reports. The APORTS Reports unit consists of the following files in the indicated directory:

*Airport.rdf* (/h/RFA/source/)  
*Airport\_Aircraft\_Cat.rdf* (/h/RFA/source/)  
*Airport\_Clearance.rdf* (/h/RFA/source/)  
*Airport\_Remark.rdf* (/h/RFA/source/)  
*Apron.rdf* (/h/RFA/source/)  
*Priority\_Geo.rdf* (/h/RFA/source/)  
*aafif\_errors.rdf* (/h/RFA/source/)  
*aports\_merge.rdf* (/h/RFA/source/)  
*aports\_xcn.rdf* (/h/RFA/source/)

#### A.2.2.7 Miscellaneous Work Package

The Miscellaneous work package has five logical units: Miscellaneous Initialization, Miscellaneous User Privilege, Miscellaneous RFADB Segmentation, Miscellaneous RFA Segmentation, and Miscellaneous Reports.

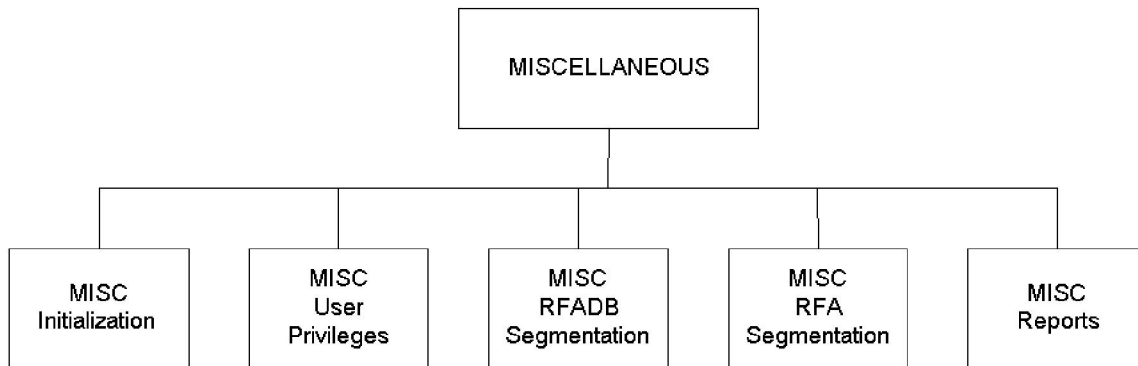


Figure A.2.2.7-1. Miscellaneous Work Package

##### A.2.2.7.1 Miscellaneous Initialization

The Miscellaneous Initialization logical unit enables RFA installer to create database objects and assign privileges to those objects that are not related to a specific reference file or that are related to two or more



specific reference files. The Miscellaneous Initialization unit consists of the following files in the indicated directories:

*alter\_user.sql* (/h/RFADB/sql/)  
*comp\_gr.sql* (/h/RFADB/sql/)  
*deinstall.sql* (/h/RFADB/sql/)  
*droptemp.sql* (/h/RFADB/sql/)  
*errors.ctl* (/h/RFADB/sql/)  
*errors.dat* (/h/RFADB/data/)  
*gen\_load.sql* (/h/RFADB/sql/)  
*gen\_pack.sql* (/h/RFADB/sql/)  
*gen\_tabs* (/h/RFADB/sql/)  
*inp\_help.par* (/h/RFADB/data/)  
*jrs\_crit.ctl* (/h/RFADB/sql/)  
*jrs\_crit.dat* (/h/RFADB/data/)  
*online\_help.exp* (/h/RFADB/data)  
*rfa\_role.sql* (/h/RFADB/sql/)  
*rfa\_ts.sql* (/h/RFADB/sql/)  
*rfa\_user.sql* (/h/RFADB/sql/)  
*role.sql* (/h/RFADB/sql/)  
*t\_m\_gr.sql* (/h/RFADB/sql/)  
*t\_m\_role.sql* (/h/RFADB/sql/)  
*temp.sql* (/h/RFADB/sql/)  
*val\_load.sql* (/h/RFADB/sql/)  
*val\_tabs.sql* (/h/RFADB/sql/)

#### **A.2.2.7.2 Miscellaneous User Privilege**

The Miscellaneous User Privilege unit allows the RFADB installer to give the required RFA privileges to a GCCS desktop user, and to revoke those privileges. The Miscellaneous User Privilege unit consists of the following files in the indicated directories:

*check\_oracle\_user.sql* (/h/RFDB/install/)  
*check\_rfa\_user.sql* (/h/RFDB/install/)  
*drop\_syn.csh* (/h/RFDB/install/)  
*make\_drop\_syn.sql* (/h/RFDB/install/)  
*make\_syn.sql* (/h/RFDB/install/)  
*new\_rfa\_user* (/h/RFDB/install/)  
*pass.csh* (/h/RFADB/Scripts/)  
*revoke\_rfa\_user* (/h/RFDB/install/)  
*syn.csh* (/h/RFDB/install/)  
*tm\_syn.sql* (/h/RFDB/install/)  
*user.sql* (/h/RFDB/install/)

#### A.2.2.7.3 Miscellaneous RFADB Segmentation

The Miscellaneous RFADB Segmentation unit enables the installer to load the RFADB segment. The Miscellaneous RFA Segmentation unit consists of the following files in the indicated directories:

- DEINSTALL* (/h/RFADB/SegDescrip/)
- Data* (/h/RFADB/SegDescrip/)
- Hardware* (/h/RFADB/SegDescrip/)
- ModName* (/h/RFADB/SegDescrip/)
- PostInstall* (/h/RFADB/SegDescrip/)
- ReleaseNotes* (/h/RFADB/SegDescrip/)
- Requires* (/h/RFADB/SegDescrip/)
- Security* (/h/RFADB/SegDescrip/)
- SegType* (/h/RFADB/SegDescrip/)
- VERSION* (/h/RFADB/SegDescrip/)
- Validated* (/h/RFADB/SegDescrip/)
- determine\_oracle\_password* (/h/RFADB/Scripts/)
- dir.csh* (/h/RFADB/Scripts/)
- dummy* (/h/RFADB/data/)
- enter\_new\_oracle\_password* (/h/RFADB/Scripts/)
- pass.csh* (/h/RFADB/Scripts/)
- warning.csh* (/h/RFADB/Scripts/)

#### A.2.2.7.4 Miscellaneous RFA Segmentation

The Miscellaneous RFA Segmentation unit enables the installer to load the RFA segment. The Miscellaneous RFA Segmentation unit consists of the following files in the indicated directories:

- DEINSTALL* (/h/RFA/SegDescrip/)
- Hardware* (/h/RFA/SegDescrip/)
- LaunchDesc.RFA* (/h/RFA/data/Profiles/)
- LaunchList.RFA* (/h/RFA/data/Profiles/)
- RFA\_launch* (/h/RFA/progs/)
- ModName* (/h/RFA/SegDescrip/)
- PostInstall* (/h/RFA/SegDescrip/)
- Profiles.RFA* (/h/RFA/data/Profiles/)
- ReleaseNotes* (/h/RFA/SegDescrip/)
- Requires* (/h/RFA/SegDescrip/)
- Security* (/h/RFA/SegDescrip/)
- SegType* (/h/RFA/SegDescrip/)
- VERSION* (/h/RFA/SegDescrip/)
- Validated* (/h/RFA/SegDescrip/)
- pass.csh* (/h/RFA/Scripts/)
- rfa.img* (/h/RFA/data/Profiles/)
- rfaenv* (/h/RFA/)

#### **A.2.2.7.5 Miscellaneous Reports**

The Miscellaneous Reports logical unit contains the report that does not relate to the other logical units. The Miscellaneous logical unit consists of the following file in the indicated directory:

*genRefFile.rdf (/h/RFA/Source/)*

#### **A.2.2.8 Software Component Interaction**

This section describes interactions of the RFA software components. The Concept of Operations section describes the high level interaction operational interaction of the RFA logical units. The logical units Interoperability section provides a detailed description of the logical units Interoperability

#### **A.2.2.9 Concept of Operations**

The RFA applications consists of two GCCS segments that must be installed at the user site. Besides installing the application software, these segments create a local ORACLE application schema, and provide access and privileges to RFA users. Once installed, the RFA application consists of a user interface developed in ORACLE Forms. The user selects a particular reference file to work with. The next step is to do a Working Copy. This step copies the “live” reference file tables from the GCCS node server’s JOPES Core database (JCDB) into RFA working tables. The user then uses the on-line or batch update processes to make changes in the working tables. As the updates are made, various reports are made available through the ORACLE Reports product which the user may view or print. Once all updates are complete, and reports verify the updates as being correct, the user selects the Network function. The Network function condenses all updates to the reference file and generates a SQL script of updates that is passed to a GCCS distribution process and is executed at each JCDB server to update the reference file across GCCS. The communication between the various modules are generally contained within the ORACLE GUI tools of Forms and Reports. However, UNIX scripts, SQL\*Plus scripts, and SQL\*Loader scripts are used to communicate with the operating system and the UNIX file system as required.

#### **A.2.2.10 Logical Units Interoperability**

Generally, logical units interact with other logical units within the same work package. Each work package will be discussed in turn.

##### **A.2.2.10.1 GEO Work Package**

The GEO work package has four logical units: GEO Initialization, GEO On-line Update, GEO Network, and GEO Reports.

The GEO Initialization logical unit enables RFA installers to create GEO database objects and grant required privileges to those objects. Modules in this unit are called during installation by the Miscellaneous RFADB Segmentation logical unit.

The GEO On-line Update is called by the Miscellaneous Forms logical unit, which contains the main menu processing of RFA. The GEO On-line Update unit stores all changes in a GEO updates data store consisting of an update table and reduced update table.

When all GEO On-line update processes have been completed, the GEO Network logical unit is called from the RFA main menu found in the Miscellaneous Forms logical unit. The GEO Network logical unit takes the data in the GEO updates data store and generates a SQL script file of differences between the local GEOGRAPHIC\_LOCATION table and the “live” JCDB table.

The GEO Reports logical unit is called directly from the user menus in the Miscellaneous Forms logical unit upon user request. In addition, the GEO On-line Update unit and the GEO Network unit call the GEO Reports unit as required.

#### **A.2.2.10.2 TUCHA Work Package**

The TUCHA work package has five logical units: TUCHA Initialization, TUCHA Batch Update, TUCHA On-line Update, TUCHA Network, and TUCHA Reports.

The TUCHA Initialization logical unit enables RFA installers to create TUCHA database objects and grant required privileges to those objects. Modules in this unit are called during installation by the Miscellaneous RFADB Segmentation logical unit.

The TUCHA Batch Update logical unit is called by the user through the Miscellaneous Forms unit. The TUCHA Batch Update unit enables RFA users to process transactions files submitted in accordance with the JRS. A record of each update is stored in a TUCHA updates data store consisting of updates tables and reduced updates tables.

The TUCHA On-line Update is called by the Miscellaneous Forms logical unit, which contains the main menu processing of RFA. The TUCHA On-line Update deletes canceled TUCHA entries via an autodelete function, and stores a record of these changes in the TUCHA updates data store.

When all TUCHA update processes have been completed, the TUCHA Network logical unit is called from the RFA main menu found in the Miscellaneous Forms logical unit. The TUCHA Network unit takes the data in the TUCHA updates data store and generates a SQL script file of differences between the local TUCHA tables and the “live” JCDB table.

The TUCHA Reports logical unit is called directly from the user menus in the Miscellaneous Forms logical unit upon user request. In addition, the TUCHA Batch unit and the TUCHA Network unit call the TUCHA Reports logical unit as required.

#### **A.2.2.10.3 TUDET Work Package**

The TUDET work package has four logical units: TUDET Initialization, TUDET Batch Update, TUDET Network, and TUDET Reports.

The TUDET Initialization logical unit enables RFA installers to create TUDET database objects and grant required privileges to those objects. Modules in this logical unit are called during installation by the Miscellaneous RFADB Segmentation logical unit.

The TUDET Batch Update is called by the Miscellaneous Forms logical unit, which contains the main menu processing of RFA. The TUDET Batch Update stores all changes in a TUDET updates data store consisting of an update table and reduced update table.

When all TUDET update processes have been completed, the TUDET Network logical unit is called from the RFA main menu found in the Miscellaneous Forms logical unit. The TUDET Network logical unit takes the data in the TUDET updates data store and generates a SQL script file of differences between the local EQUIPMENT\_TYPE table and the “live” JCDB table.

The TUDET Reports logical unit is called directly from the user menus in the Miscellaneous Forms logical unit upon user request. In addition, the TUDET Batch Update unit and the TUDET Network unit call the TUDET Reports logical unit as required.

#### **A.2.2.10.4 LFF Work Package**

The LFF work package has four logical units: LFF Initialization, LFF Batch Update, LFF Network, and LFF Reports.

The LFF Initialization logical unit enables RFA installers to create LFF database objects and grant required privileges to those objects. Modules in this logical unit are called during installation by the Miscellaneous RFADB Segmentation logical unit.

The LFF Batch Update is called by the Miscellaneous Forms logical unit, which contains the main menu processing of RFA. The Batch Update stores all changes in a LFF updates data store consisting of an update table and reduced update table.

When all LFF update processes have been completed, the LFF Network logical unit is called from the RFA main menu found in the Miscellaneous Forms logical unit. The LFF Network logical unit takes the data in the LFF updates data store and generates a SQL script file of differences between the local LFF tables and the “live” JCDB tables.

The LFF Reports logical unit is called directly from the user menus in the Miscellaneous Forms logical unit upon user request. In addition, the LFF Batch Update unit and the LFF Network unit call the LFF Reports logical unit as required.

#### **A.2.2.10.5 PORTS Work Package**

The PORTS work package has four logical units: PORTS Initialization, PORTS Batch Update, PORTS Network, and PORTS Reports.

The PORTS Initialization logical unit enables RFA installers to create PORTS database objects and grant required privileges to those objects. Modules in this logical unit are called during installation by the Miscellaneous RFADB Segmentation logical unit.

The PORTS Batch Update is called by the Miscellaneous Forms logical unit, which contains the main menu processing of RFA. The Batch Update stores all changes in a PORTS updates data store consisting of an update table and reduced update table.

When all PORTS update processes have been completed, the PORTS Network logical unit is called from the RFA main menu found in the Miscellaneous Forms logical unit. The PORTS Network logical unit takes the data in the PORTS updates data store and generates a SQL script file of differences between the local PORTS tables and the “live” JCDB tables.

The PORTS Reports logical unit is called directly from the user menus in the Miscellaneous Forms logical unit upon user request. In addition, the PORTS Batch Update unit and the PORTS Network unit call the PORTS Reports logical unit as required.

#### **A.2.2.10.6 APORTS Work Package**

The APORTS work package has four logical units: APORTS Initialization, APORTS Batch Update, APORTS Network, and APORTS Reports.

The APORTS Initialization logical unit enables RFA installers to create APORTS database objects and grant required privileges to those objects. Modules in this logical unit are called during installation by the Miscellaneous RFADB Segmentation logical unit.

The APORTS Batch Update is called by the Miscellaneous Forms logical unit, which contains the main menu processing of RFA. The Batch Update stores all changes in a APORTS updates data store consisting of an update table and reduced update table.

When all APORTS update processes have been completed, the APORTS Network logical unit is called from the RFA main menu found in the Miscellaneous Forms logical unit. The APORTS Network logical unit takes the data in the APORTS updates data store and generates a SQL script file of differences between the local APORTS tables and the “live” JCDB tables.

The APORTS Reports logical unit is called directly from the user menus in the Miscellaneous Forms logical unit upon user request. In addition, the APORTS Batch Update unit and the APORTS Network unit call the APORTS Reports logical unit as required.

#### **A.2.2.10.7 Miscellaneous Package**

The Miscellaneous RFA segmentation logical unit is initiated by the RFA installer and loads all the application software to the UNIX application server or client. This includes the modules for each logical unit except GEO Initialization, TUCHA Initialization, TUDET Initialization, LFF Initialization, PORTS Initialization, and APORTS Initialization.

The Miscellaneous RFADB Segment is initiated by the RFA installer logical unit loads the GEO Initialization unit, the TUCHA Initialization unit, the TUDET Initialization unit, the LFF Initialization, the PORTS Initialization unit, and the APORTS initialization unit to the database server. The Miscellaneous RFADB segment then calls each of the other work package initialization units.

The Miscellaneous Forms logical unit is called directly by the RFA user. This unit has the RFA main menu which allows the user to call the GEO On-line Update unit, the GEO Network unit, the GEO Reports unit, the TUCHA Batch Update unit, the TUCHA Online Update unit, the TUCHA Network unit, the TUCHA Reports unit, the TUDET Batch Update, the TUDET Network unit, the LFF Batch Update unit, the LFF Network, the LFF Reports unit, the PORTS Batch Update unit, the PORTS Network, the PORTS Reports unit, the APORTS Batch Update unit, the APORTS Network, the APORTS Reports unit, and the Miscellaneous Reports unit.

The Miscellaneous Reports unit contains all reports that do not relate to the other logical units. This unit is called at user request from the Miscellaneous Forms unit.

### A.2.3 Environmental Variables

RFA uses four environmental variables to point to various directories. Two of the variables are used directly by RFA:

*RFA\_HOME* -- defines the directory path where the RFA scripts are located.

*RFA\_NET* -- defines the directory path where external input files are expected to be found, and where RFA puts files that need to be exported by the user.

*RFA\_FILES* -- defines the directory path where RFA output scripts are loaded.

RFA also sets two environmental variables used by ORACLE desktop tools:

*ORACLE\_PATH* -- defines the directory where ORACLE Forms 4.0 will find object forms.

*REP\_PATH* -- defines the directory where ORACLE Reports 2.0 will find objects reports.

These environmental variables are set in */h/RFA/rfaenv* which is sourced by the RFA launch program */h/RFA/progs/RFA\_launch* and */h/RFA/progs/RFA\_maint*.

### A.2.4 Database Variables

RFA creates several ORACLE database objects. RFA creates users, tablespace, roles, tables, sequences, database triggers, packages, constraints and indexes. RFA users create private synonyms. Files that create or control these objects are described below.

#### A.2.4.1 RFA User, Tablespace, and Role

*rfa\_ts.sql* (/h/RFADB/sql/)  
*rfa\_user.sql* (/h/RFADB/sql/)  
*role.sql* (/h/RFADB/sql/)

#### A.2.4.2 RFA Tables and Sequences

*apo\_tabs* (/h/RFADB/sql/)  
*gen\_tabs.sql* (/h/RFADB/sql/)  
*geo\_tabs.sql* (/h/RFADB/sql/)  
*lff\_tabs* (/h/RFADB/sql/)  
*por\_tabs* (/h/RFADB/sql/)  
*tuc\_tabs* (/h/RFADB/sql/)  
*tud\_tabs* (/h/RFADB/sql/)

#### **A.2.4.3 RFA Database Triggers**

*apo\_tgrs.sql* (/h/RFADB/sql/)  
*geo\_tgrs.sql* (/h/RFADB/sql/)  
*lff\_tgrs.sql* (/h/RFADB/sql/)  
*por\_tgrs.sql* (/h/RFADB/sql/)  
*tuc\_tgrs.sql* (/h/RFADB/sql/)  
*tud\_tgrs.sql* (/h/RFADB/sql/)

#### **A.2.4.4 RFA Database Packages**

*gen\_pack* (/h/RFADB/sql/)  
*lff\_bods.sql* (/h/RFADB/sql/)  
*lff\_bulb.sql* (/h/RFADB/sql/)  
*lff\_buls.sql* (/h/RFADB/sql/)  
*lff\_spcs.sql* (/h/RFADB/sql/)  
*por\_basp.sql* (/h/RFADB/sql/)  
*por\_bd1b.sql* (/h/RFADB/sql/)  
*por\_bd2b.sql* (/h/RFADB/sql/)  
*por\_bd3b.sql* (/h/RFADB/sql/)  
*por\_bd4b.sql* (/h/RFADB/sql/)  
*por\_bt1b.sql* (/h/RFADB/sql/)  
*por\_bt2b.sql* (/h/RFADB/sql/)  
*por\_bt3b.sql* (/h/RFADB/sql/)  
*por\_bt4b.sql* (/h/RFADB/sql/)  
*por\_btpb.sql* (/h/RFADB/sql/)  
*por\_cnob.sql* (/h/RFADB/sql/)  
*por\_conx.sql* (/h/RFADB/sql/)  
*por\_edb1.sql* (/h/RFADB/sql/)  
*por\_edb2.sql* (/h/RFADB/sql/)  
*por\_edb3.sql* (/h/RFADB/sql/)  
*por\_edb4.sql* (/h/RFADB/sql/)  
*por\_edsp.sql* (/h/RFADB/sql/)  
*por\_popb.sql* (/h/RFADB/sql/)  
*por\_pope.sql* (/h/RFADB/sql/)  
*por\_pops.sql* (/h/RFADB/sql/)  
*por\_preb.sql* (/h/RFADB/sql/)  
*por\_roll.sql* (/h/RFADB/sql/)  
*tuc\_pack.sql* (/h/RFADB/sql/)  
*tuc\_rpts.sql* (/h/RFADB/sql/)  
*tud\_pack.sql* (/h/RFADB/sql/)

#### **A.2.4.5 RFA Constraints and Indexes**

*apo\_conx.sql* (/h/RFADB/sql/)  
*geo\_conx.sql* (/h/RFADB/sql/)  
*lff\_conx.sql* (/h/RFADB/sql/)  
*por\_conx.sql* (/h/RFADB/sql/)



*tuc\_conx.sql* (/h/RFADB/sql/)  
*tud\_conx.sql* (/h/RFADB/sql/)

#### **A.2.4.6 RFA Privileges**

*comp\_gr.sql* (/h/RFADB/sql/)  
*t\_m\_gr.sql* (/h/RFADB/sql/)

#### **A.2.4.7 User Privileges**

*alter\_user.sql* (/h/RFADB/sql/)  
*check\_oracle\_user.sql* (/h/RFADB/install/)  
*make\_drop\_syn.sql* (/h/RFADB/install/)  
*make\_syn.sql* (/h/RFADB/install/)  
*revoke\_user.sql* (/h/RFADB/install/)  
*rfa\_role.sql* (/h/RFADB/sql)  
*t\_m\_role.sql* (/h/RFADB/sql)  
*tm\_syn.sql* (/h/RFADB/install/)  
*user.sql* (/h/RFADB/install/)

### **A.2.5 External System Interfaces**

RFA receives input files from Services and agencies via the JRS. The location of these files is controlled by the RFA\_NET environmental variable. RFA\_NET points to a directory called rfa\_net which is subordinate to the users home directory (~/.rfa\_net). In the Network process, RFA provides output files in the same directory. GEO, TUCHA, and TUDET provide ASCII JRS transaction files that can be transferred to the Top Secret Support System (TS3) to match their reference files with GCCS. This directory is also used to hold ASCII SQL script files that are passed to the GCCS network distribution function to update reference files at all GCCS JCDB database nodes.

## A.3 Compilation/Build Procedures

### A.3.1 Forms Object Files

To build the RFA forms object files (indicated by an extension of *.fmx*), perform the following procedure:

1. Change the default directory to `/h/RFA/source`.
2. Enter: `${ORACLE_HOME}/bin/f40desm form_name.fmb`
3. Click on **{OK}** in the Canvas-Views window.
4. From any Forms window, click **{File}** and then click **{Generate As}**.
5. In the generate box enter: `/h/RFA/rfa_home/form_name.fmx` and click **{OK}**. If the *.fmx* file exists, you will be asked whether you want to overwrite it. Answer in the affirmative.
6. Exit ORACLE Forms by clicking **{File}** on the menu bar and then clicking **{Quit}**.

### A.3.2 Report Object Files

To build the RFA report object files (indicated by an extension of *.rep*), perform the following procedure:

1. Change the default directory to `/h/RFA/source`.
2. Enter: `${ORACLE_HOME}/bin/r20desm report_name.rdf`
3. From any report writer window, click **{File}** and then click **{Generate As}**.
4. In the generate box, enter `/h/RFA/rfa_home/report_name.rep` and click **{OK}**. If the *.rep* file exists, you will be asked whether you want to overwrite it. Answer in the affirmative.
5. Exit ORACLE Reports by clicking **{File}** on the menu bar and then clicking **{Quit}**.

## **A.4 Modification Procedures**

This section explains how RFA should be modified following receipt of a Global Software Problem Report (GSPR). The following subsections specify information necessary to perform modifications.

### **A.4.1 Supporting Software**

The following software packages are required for RFA use:

- a. ORACLE RDBMS 7 Server Release 7.1.4.1.0 w/distributed option
- b. ORACLE PL/SQL Release 2.1.4.0.0
- c. ORACLE Forms 4.0.13.20.0
- d. ORACLE Reports 2.0.14.6.2
- e. ORACLE SQL\*Loader Release 7.1.4.1.0
- f. ORACLE SQL\*Plus Release 3.1.3.5.1
- g. SQL\*NET ver 2.0.15.0.0
- h. JOPES Core Database (SMDB Segment)
- i. Solaris 2.3/2.4

### **A.4.2 Databases/Data Files**

RFA creates many ORACLE objects during the RFADB segment installation. The files to create these objects have the extension *.sql* and are located in /h/RFADB/sql and /h/RFADB/install. All *.sql* files should create a log file in the /tmp directory. SQL scripts that create database packages should create a log file ending in *\_pack.log*. SQL scripts that create database triggers should create log files that end in *\_trgs.log*. SQL scripts that create constraints should create log files ending with *\_conx.log*. Procedures for evaluating the log files to determine if the installation was successful should be documented in the /h/RFADB/SegDescrip/ReleaseNotes file.

### **A.4.3 Design, Coding and Other Conventions**

The following design and coding standards and procedures apply to modification efforts.

#### **A.4.3.1 Coding Standards**

RFA was created in accordance with the following:

- a. Language-independent (SQL scripts),
- b. RFA ORACLE Forms 4.0,
- c. RFA ORACLE Reports 2.0,
- d. Stored Packages and Procedures, and
- e. UNIX scripts.

These standards appear in JOPES Software Development Standards, and have been informally provided to the Government.

#### **A.4.3.2 Applicable Procedures**

The procedures in the following subsections should be used to modify each type of source code file found in RFA.

#### **A.4.3.3 ORACLE Forms 4.0 Software**

To update the RFA screens, the programmer must execute the ORACLE Forms 4.0 designer. The programmer must first make sure the ORACLE environment variables are set. Then, change to the directory path to the programmers working area where the source form is located. At the prompt, enter: **f40desm**. For details on using the ORACLE Forms designer, see ORACLE Forms User's Guide and ORACLE Forms Reference Manual. The programmer will need to connect to the ORACLE database as user RFA before generating the executable. When testing the executable, the programmer should log on to ORACLE as a user account that has been granted the RFA\_USER role. Testing should not be done as the ORACLE user RFA. Prologues for all forms are found at the Form Module Comments icon. When the source form is "checked in" to RFA CM, the source is placed in /h/RFA/source, the form object is re-generated under /h/RFA/rfa\_home. Source files have a *.fmb* extension and object files have a *.fmx* extension.

#### **A.4.3.4 ORACLE Reports 2.0 Software**

To update the RFA reports, the programmer must execute the ORACLE Reports 2.0 designer. The programmer must first make sure the ORACLE environment variables are set. Then, change to the directory path to the programmers working area where the source report is located. At the prompt, enter '**r20desm**'. For details on using the ORACLE Reports designer, see Building Reports with ORACLE Reports, and ORACLE Reports Reference Manual. The programmer will need to connect to the ORACLE database as user RFA before generating the executable. When testing the executable, the programmer should log on to ORACLE as a user account that has been granted the RFA\_USER role. Testing should not be done as the ORACLE user RFA. Prologues for all forms are found at the Reports Global Properties Comments icon. When the programmer "checks" in the form, RFA CM will move the source form to /h/RFA/source, re-generate the object file under the RFA ORACLE user and place the object file in /h/RFA/rfa\_home for integration testing. Source files have a *.rdf* extension and object files have a *.rep* extension.

#### **A.4.3.5 SQL Scripts**

The SQL scripts are ASCII files generally used in the used in segment loading. As such they reside in the /h/RFADB/install and /h/RFADB/sql directories. There are two *.sql* files in /h/RFA/rfa\_home that are used to transfer data from ORACLE tables to UNIX ASCII. All files have *.sql* as the extension.

#### **A.4.3.6 CTL Scripts**

The Control File (CTL) scripts are ASCII files used for loading ASCII text files into RFA tables. These files reside in directory /h/RFADB/sql for the segment loading process and in /h/RFA/rfa\_home for use in the RFA application. All files have *.ctl* as the extension.

#### **A.4.3.7 UNIX Scripts**

UNIX scripts are ASCII files used to interface with the operating system. Most UNIX script files are used in the installation of the RFA segments. UNIX scripts either have no extension or have the extension of *.csh*.

## **A.4.4 Compilation/Build Procedures**

### **A.4.4.1 Forms Object Files**

To build the RFA forms object files (indicated by an extension of *.fmx*), perform the following procedure:

1. Change the default directory to */h/RFA/source*.
2. Enter: **`${ORACLE_HOME}/bin/f40desm form_name.fmb`**
3. Click on **{OK}** in the Canvas-Views window.
4. From any Forms window, click **{File}** and then click **{Generate As}**.
5. In the generate box enter: **`/h/RFA/rfa_home/form_name.fmx`** and click **{OK}**. If the *.fmx* file exists, you will be asked whether you want to overwrite it. Answer in the affirmative.
6. Exit ORACLE Forms by clicking **{File}** on the menu bar and then clicking **{Quit}**.

### **A.4.4.2 Report Object Files**

To build the RFA report object files (indicated by an extension of *.rep*), perform the following procedure:

1. Change the default directory to */h/RFA/source*.
2. Enter: **`${ORACLE_HOME}/bin/r20desm report_name.rdf`**
3. From any report writer window, click **{File}** and then click **{Generate As}**.
4. In the generate box, enter **`/h/RFA/rfa_home/report_name.rep`** and click **{OK}**. If the *.rep* file exists, you will be asked whether you want to overwrite it. Answer in the affirmative.
5. Exit ORACLE Reports by clicking **{File}** on the menu bar and then clicking **{Quit}**.

#### **A.4.5 Integration and Testing Procedures**

To perform integration and test of RFA modification, the RFA system must be properly initiated. On a GCCS client workstation, click on the **{RFA icon}** on the GCCS desktop, or select RFA from the JNAV system. Follow the predetermined test plan and compare the results with those expected.

## **A.5      Qualification Provisions**

The RFA application is delivered on magnetic tape media, using the standard UNIX "tar" format. The current version identification can be verified through manual examination of the version files */h/RFA/SegDescrip/VERSION* and */h/RFADB/SegDescrip/VERSION*.